Minimizing Calibrated Loss using Stochastic Low-Rank Newton Descent for large scale image classification

Wafa Bel Haj Ali, Michel Barlaud, Richard Nock

Technical report

Minimizing Calibrated Loss using Stochastic Low-Rank Newton Descent for large scale image classification

Wafa BelHajAli, Richard Nock, Michel Barlaud

April 18, 2013

Contents

Abstract

A standard approach for large scale image classification involves high dimensional features and Stochastic Gradient Descent algorithm (SGD) for the minimization of classical Hinge Loss in the primal space. Although complexity of Stochastic Gradient Descent is linear with the number of samples these method suffers from slow convergence. In order to cope with this issue, we propose here a Stochastic Low-Rank Newton Descent (SLND) for minimization of any calibrated loss in the primal space. SLND approximates the inverse Hessian by the best low-rank approximation according to squared Frobenius norm. We provide core optimization for fast convergence. Theoretically speaking, we show explicit convergence rates of the algorithm using these calibrated losses, which in addition provide working sets of parameters for experiments. Experiments are provided on the SUN, Caltech256 and ImageNet databases, with simple, uniform and efficient ways to tune remaining SLND parameters. On each of these databases, SLND challenges the accuracy of SGD with a speed of convergence faster by order of magnitude.

1 Introduction

Large scale image classification requires computational efficiency. To cope with these issues, current standard approaches involves involves high dimensional features like Fischer Vectors [16] or super vectors and Support Vector Machines (SVM) with linear kernels for training [21].

The classical approach introducing SVM first state dual formulation [19] where the task is to minimize empirical loss with a regularization term.

The first alternative approach on primal optimization [11] used conjugate gradient or cutting plane algorithms [9].

Recent state of the art papers focus on the more efficient stochastic gradient descent algorithm SGD[24, 5] , the "PEGASOS" algorithm [18], with linear complexity in the number of samples.

Although SGDdescent methods perform as well as batch solvers at a fraction of cost, they still suffers from slow convergence. Two approaches were recently proposed in order to cope with this issue. The first is the natural gradient approach, which incorporates the estimation of the Riemannian metric tensor using Fisher information [1].

The second alternative approaches are based on a stochastic version of the quasi Newton Broyden-Fletcher-Golfarb-Shanno (BFGS) optimization algorithm. The first one is a low memory stochastic version of the BFGS quasi Newton method [17]. Although their oBFGS method reduces the number of iterations, each iteration requires a multiplication by a low rank matrix. Unfortunately this computational complexity is often larger than the gains associated with the quasi-Newton

update as pointed in [3]. In order to cope with this complexity [3, 4] proposed a "SGD-QN" algorithm with an update using the diagonal of the Hessian matrix. Unfortunately there are no proof of convergence of their "SGD-QN" algorithm.

Our high-level contribution is a new stochastic Low-Rank Newton scheme and experimental validations on three large and challenging domains: SUN and Caltech256 and ImageNet. To be more specific, the novelty of our paper includes:

(i) a new Stochastic Newton descent algorithm, SLND, which approximates the inverse Hessian by a low-rank approximation which we prove is the best according to the squared Frobenius norm. SLND minimizes any *classification calibrated risk*, that may ensure convergence towards Bayes rule;

(ii) the proof of convergence of SLND, which provides rates of convergence and working set of parameters for the experiments, including the step size parameter η_t, typically in the order $\Omega(1/m)$;

(iii) experimental results display that SLND has linear complexity both in term of the number of samples and the dimension of the features and challenges the accuracy of SGDwhile being a magnitude faster.

The remaining of the paper is organized as follows: Section 2 presents calibrated risks, Section 3 provides our new algorithm SLND with several key steps for its core optimization. Section 4 presents experiments on large data sets, Section 5 presents convergence proof of our new algorithm SLND and Section 6 conclude the paper.

2 Calibrated risks

We first provide some definitions. Our setting is multiclass, multilabel classification. We have access to an input set of m examples (or prototypes, or samples), $\mathcal{S} \doteq \{(\boldsymbol{x}_i, \boldsymbol{y}_i), i = 1, 2, ..., m\}$. Vector $\boldsymbol{y}_i \in \{-1, +1\}^C$ encodes class memberships, assuming $y_{ic} = +1$ means that observation $\boldsymbol{x}_i$ belongs to class c. A classifier h is a function mapping observations to real-valued vectors in $\mathbb{R}^C$. Given some observation $\boldsymbol{x}$, the sign of coordinate c in $h(\boldsymbol{x})$, h_c, gives whether h predicts that $\boldsymbol{x}$ belongs to class c, while its absolute value may be viewed as a confidence in classification.

To learn this classifier h, we focus on the minimization of a total risk which

sums per-class losses:

$$\varepsilon_{\mathrm{F}}(h, \mathcal{S}) \doteq \frac{1}{C} \sum_{c=1}^{C} \underbrace{\frac{1}{m} \sum_{i=1}^{m} \mathrm{F}(y_{ic} h_c(\boldsymbol{x}_i))}_{\varepsilon_{\mathrm{F}}(h_c, \mathcal{S})} . \tag{1}$$

Recent advances in classification allow to precisely define constraints with whom such losses have to comply, to meet statistical and computational properties particularly desirable in handling large, complex and noisy classification problems [2, 14, 20]. There are three constraints: F is convex, differentiable and meets $F(x) = -x + \int f$, where $f : \mathbb{R} \to [0, 1]$ is increasing and symmetric with respect to $(0, 1/2 = f(0))$. Details are out of the scope of this paper, but the fundamental intuition is that f directly maps a real valued prediction h_c to a posterior estimation for class c. This last constraint ensures that the loss at hand F is Fisher consistent and proper, properties with which convenient form of convergence to Bayes rule are accessible through minimizing (1). We call losses that meet these constraints, and the total risks by extension, as *classification calibrated*. Examples of classification calibrated losses include the squared and the logistic losses. In this paper, we first consider the logistic loss:

$$\mathrm{F}^{log}(x) \doteq \ln(1 + \exp(-x)) . \tag{2}$$

We also consider a classification calibrated version of the popular but not differentiable Hinge loss ($hinge(x) \doteq \max\{0, -x\}$, proof omitted):

$$\mathrm{F}^{hinge}(x) \doteq hinge(x) - \ln(2 + |x|) . \tag{3}$$

Fig 1 shows the logistic loss and the calibrated Hinge loss. We also plot Hinge loss and the exponential loss for comparison. Note that $\mathrm{F}''(x) \leq \mathrm{F}''(0)$ for the calibrated losses (2) and (3).

Remark: there is no regularization term in (1), which is quite non-standard if we refer to the classical SVM or SGD approaches [3]. In fact, the iterative minimization we carry out for (1) explicitly integrates a "sparsity" term in the form of low rank updates of the classifier h.

3 SLND: Stochastic Low-Rank Newton Descent

3.1 Computing gradient update

To carry out the minimization of (1), we adopt a mainstream 1-vs-rest training scheme which is more efficient among different approaches [15, 22]. For each class

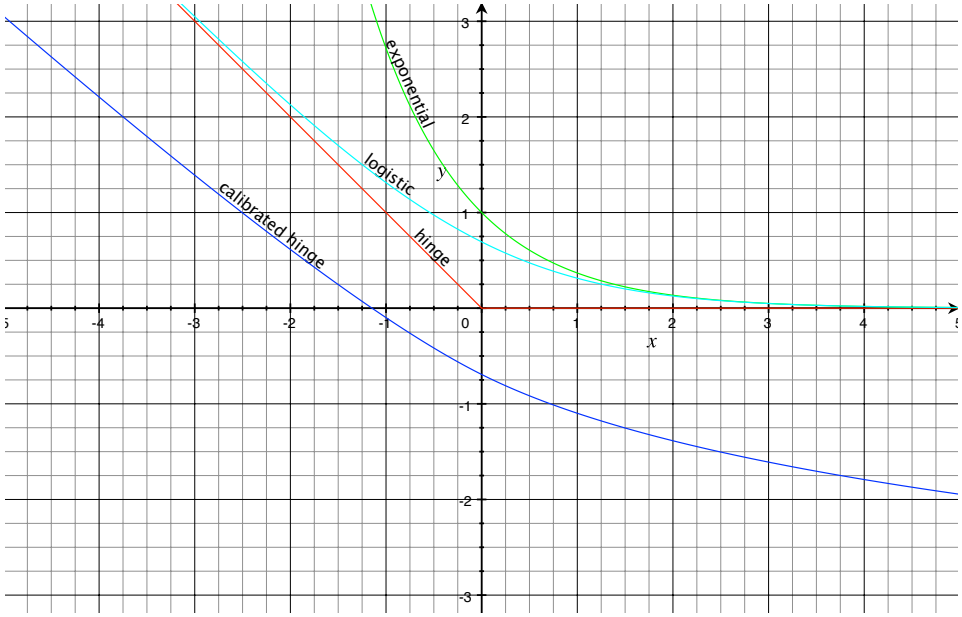

Figure 1: Calibrated losses F: the logistic and calibrated Hinge losses considered in this paper.

$c = 1, 2, ..., C$, we carry out separately the minimization of $\varepsilon_{\mathrm{F}}(h_c, \mathcal{S})$ in $\varepsilon_{\mathrm{F}}(h, \mathcal{S})$. To do so, it fits the c^{th} component of w by considering the two-class problem of class c versus all others. In what follows, we thus drop c to simplify notations.

In this paper we focus on the classical linear classifier defined as $h(x_i) = w^\top x_i$. The Goal is to learn w for each class $c = 1, 2, ..., C$ minimizing the following criterion:

$$\varepsilon_{\mathrm{F}}(w, \mathcal{S}) \quad \doteq \quad \frac{1}{m} \sum_{i=1}^{m} \mathrm{F}(y_{ic} w^\top x_i) \ . \tag{4}$$

To approximate the optimal w^*, we carry out stochastic Newton updates of a current w, noted w_t. At iteration t, we pick randomly a sample $x_i \in \mathcal{S}$ and perform the update:

$$w_{t+1} = w_t - \eta_t \underbrace{\left(\frac{\partial^2 \varepsilon_{\mathrm{F}}(w_t, x_i)}{\partial^2 w_t} \right)^{-1} \frac{\partial \varepsilon_{\mathrm{F}}(w_t, x_i)}{\partial w_t}}_{\Delta w_t} \ , \tag{5}$$

where $\eta_t > 0$ controls the strength of the update, the first derivative or the gradient ∇ is:

$$\frac{\partial \varepsilon_{\mathrm{F}}(w_t, x_i)}{\partial w_t} = y_i \mathrm{F}' \left(y_i w_t^T x_i \right) x_i \ , \tag{6}$$

and the second derivative, or the Hessian $\mathcal{H}$, is:

$$\frac{\partial^2 \varepsilon_{\mathrm{F}}(w_t, x_i)}{\partial^2 w_t} = \mathrm{F}'' \left(y_i w_t^T x_i \right) x_i x_i^T \ . \tag{7}$$

Computing the inverse of such a Gram matrix is an ill posed problem as the rank of $\mathcal{H}$ is one. Computing a pseudo-inverse is possible but on such pointwise estimates, the effect of noise in data can be dramatic for generalization [3]. We circumvent these problems by first replacing $\mathcal{H}$ by its average over a subset of $m' \leq m$ examples, which increases its rank. $\mathcal{H}$ becomes an estimation of the covariant matrix computed over the subset of examples. Then, for some typically small $k > 0$, we compute a low-rank pseudo-inverse, *i.e.* a low-rank approximation of its inverse, $\mathcal{H}^*$, as follows, where $rank(\mathcal{H}') = k$ is user-fixed. First, we perform a diagonalization of $\mathcal{H} = \mathcal{P}\mathcal{D}\mathcal{P}^\top$ where (non-negative) diagonal values are ordered in decreasing order, $d_{11} \geq d_{22} \geq ... \geq d_{uu} = 0 = ...d_{nn}$, where $u \geq k$. Denote $\mathcal{P}_{|k}$ the $m \times k$ matrix containing the first k columns of $\mathcal{P}$, and resp. $\mathcal{D}_{|k}$ as the $k \times k$ diagonal matrix of their eigenvalues. We finally compute $\mathcal{H}^*$:

$$\mathcal{H}^* = \mathcal{P}_{|k} \mathcal{D}_{|k}^{-1} \mathcal{P}_{|k}^\top \ . \tag{8}$$

The update (5) becomes:

$$w_{t+1} = w_t - \eta_t y_i F' \left(y_i w_t^T x_i \right) \mathcal{H}^* x_i \ . \tag{9}$$

3.2 Core optimization

Since we use 1-vs-rest training scheme, the training set is usually highly unbalanced when the number of class increases, examples not in class c outnumbering those in class c, for any c. When class c is a minority class, this is even more dramatic. To dampen the negative consequences, we follow the sampling balancing approach proposed by [15]. When learning class c against the rest, we use all examples from class c (the positives), while sampling a subset of the rest of the other classes (the negatives) of the same size.

Furthermore, in order to optimize computational complexity, once $\mathcal{H}^*$ is computed, we precompute for all the training set a weighted preprocessing of the features:

$$x_i^* = \mathcal{H}^* x_i \ . \tag{10}$$

Notice that this is done only once for a given $\mathcal{H}^*$. This saves significant training time and the computational complexity of each iteration in SLND is basically of the same order as classical SGD [3]. The final update in SLND is:

$$w_{t+1} = w_t - \eta_t y_i F' \left(y_i w_t^T x_i \right) x_i^* \ . \tag{11}$$

Finally, the tuning of η_t is a non-trivial problem for gradient or Newton approaches [3]. In general, small positive values are chosen but little can be said as to whether convergence holds, and if so, under which rates. We prove an explicit convergence rate for SLND in Theorem 2 hereafter which provides us with expressions for η_t typically in the order $\Omega(1/m)$ and $O(1/\sqrt{m})$. The values we have chosen in our implementation of SLND belong to this range and are thus compatible with the formal convergence rates shown for SLND.

3.3 Remarks

There are several comparisons to make about SLND with respect to other prominent approaches. First, SLND is not related to (linear) SVM, as there is no regularization term in the criterion (4), which explains the difference between the right hand-side term in w_t in (4) and the term in $(1 - \lambda)w_t$ which would follow from the classical linear SVM cost function, where λ controls the strength of regularization [3]. Also, SLND is significantly different from dimensionality reduction techniques like PCA or general non-linear manifold learning, which would carry out dimensionality reduction as a preconditioning *on data* and on w, thus working on the reduced domain. Notice also that (10) is not a preconditioning of data, as each iteration in (11) makes use of both x_i and x_i^*. In addition, SLND is also different from the quasi newton (L)BFGS family [13] [17] as the approximation to the Hessian inverse is carried out in a different way. Moreover SLND differs from quasi-Newton methods for SVM [3] since we do not restrict the Hessian approximation to be diagonal (thus omitting all covariance terms). Finally, SLND is not a natural gradient approach (which incorporates Riemannian metric tensor [1]) and thus SLND does not require the computation of the Fisher information matrix.

4 Experimental evaluation

4.1 Settings

We mainly report and discuss experiments of SLND versus SGD which represents the state of art among the classifiers on large scale datasets [24, 5] , [18], [15].

We use Fisher vectors (FV) [16] as efficient **features** to represent images. Fisher Vectors are computed over densely extracted SIFT descriptors (FV_s) and local color features (FV_{sc}), both projected with PCA in a subspace of dimension 64. Since the goal of the paper is to compare SLND versus SGDwe use Fisher Vectors using a vocabulary of only 16 Gaussian to limit memory requirement. Each Fisher Vectors are normalized separately for both channels and then combined by concatenating the two features vectors (FV_{s+sc}). This approach leads to to a 4K dimensional features vector.

We report experimental results on three **datasets**, Caltech256, SUN and ImageNet which are among the most challenging datasets publicly available for large scale image classification:

- Caltech256 [8]: This dataset is a collection of 30607 images of 256 object classes. Following classical evaluation, we use 30 images/class for training and the rest for testing.
- SUN [23]: This dataset is a collection of 108656 images divided into 397 scenes categories. We set the number of training images per class to 50 and we test on the remaining.
- ImageNet [6]: We use the dataset of the ImageNet Large Scale Visual Recognition Challenge 2010 (ILSVRC2010)[1] with its 1000 categories. It provides $1.2M$ of images for training step and $150K$ for testing.

To **compare** algorithms, we use top1 and top5 accuracies (ACC), defined respectively as the proportion of examples that was correctly labelled and the proportion of those for which the correct class belongs to the top5 predicted images [12]. We first analyse parameter of SLND and then the convergence of SLND.

4.2 Tuning parameters of SLND

Our algorithm requires the tuning of only three parameters: the step size parameter η_t , the rank k and the number of sample m' for the computation of the covariance matrix. The step size parameter η_t is typically in the order $\Omega(1/m)$.
Let us study the influence of parameters k and m'.

[1] http://image-net.org/challenges/LSVRC/2010/index

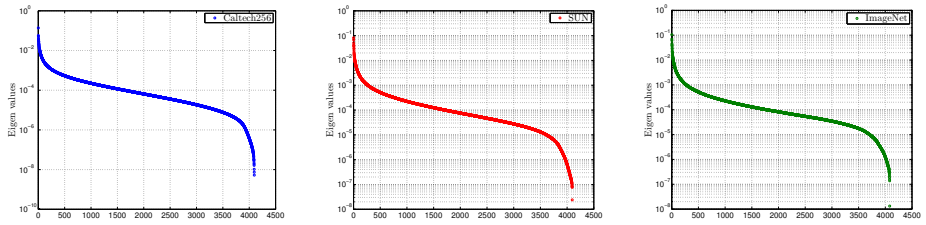

Figure 2: Eigenvalues of the covariance matrix on Caltech256 (left), SUN (center) and ImageNet (right).

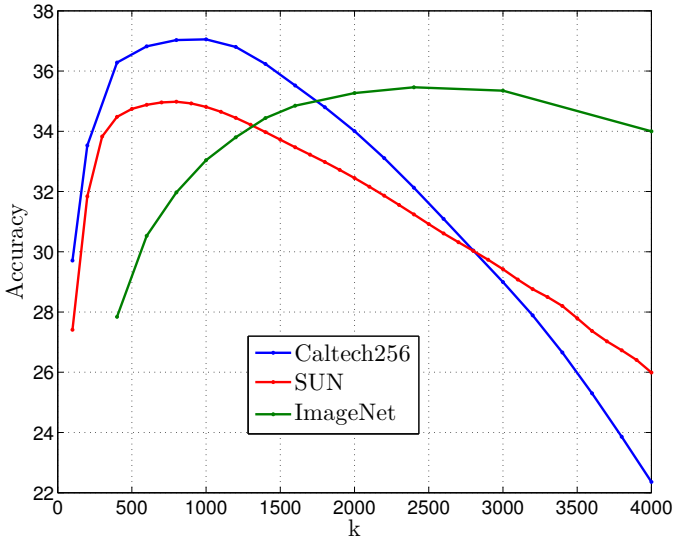

Figure 3: Accuracy as a function of the rank of the Hessian matrix on Caltech256 (blue), SUN (red) and ImageNet (green).

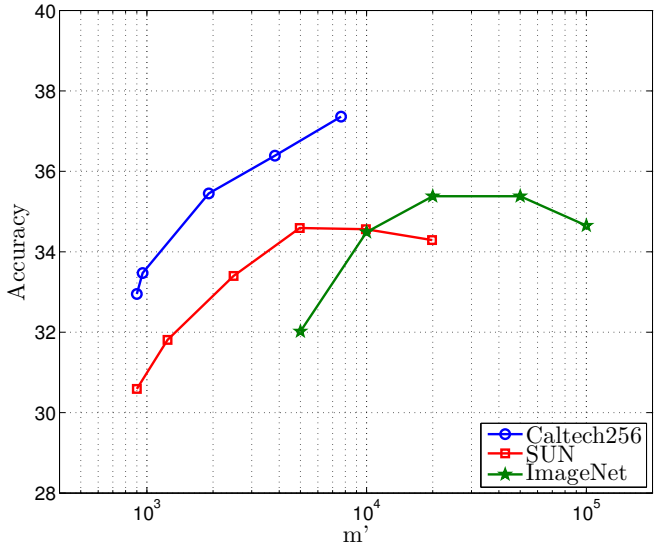

Figure 4: Accuracy as a function of the number of samples used for the computation of the Hessian matrix on Caltech256 (blue), SUN (red) and ImageNet (green, see text).

Fig 2 shows the eigenvalues of the covariance matrix, ordered from the largest to the smallest. All curves have the same sigmoid shape, and our choices of k ensure that we get all the significantly large eigenvalues. Recall that although the covariance matrix is positive-definite, the condition number is very large resulting in an ill-conditioned problem.

In order to cope with this issue, we study the accuracy as a function of the rank of the inverse of the Hessian: Fig 3 shows that accuracy curve has its max in a large rank plateau, and furthermore this plateau is similar regardless of the domain. Fig 4 shows the accuracy as a function of samples m' used for computing the covariance matrix. Fluctuations of m' imply fluctuations in the accuracy, but the range of the accuracy is not very large for reasonable values of m'.

To summarize, the eigenvalues curve, the curve accurary as a function of the rank k and to a lesser extent the curve accuracy as a function of m' have the same behavior for all databases. Thus, based on the above-experiments, both rank k and m' in SLND can be quite easily tuned.

4.3 Convergence rate analysis

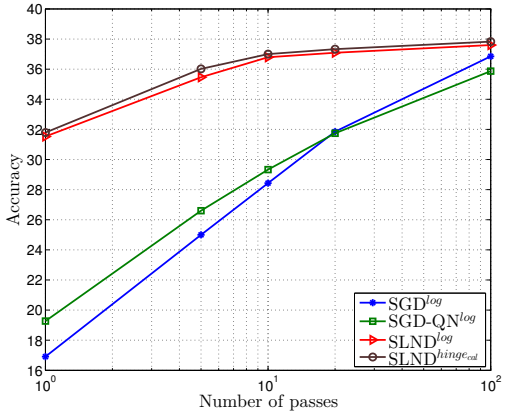

Figure 5: Top1 accuracies as a function of number of passes (iterations / skips) for SGD and SLND on Caltech256

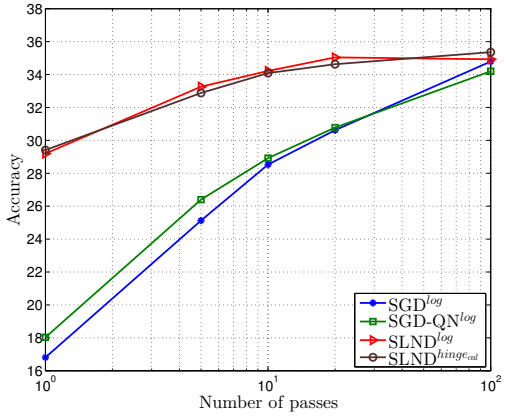

Figure 6: Top1 accuracies as a function of number of passes (iterations / skips) for SGD and SLND on SUN.

Training time and convergence of algorithms are very important for large scale data base processing.
We plot on fig 5 and 6 the convergence of SGD with logistic loss, SLND both

for Logistic Loss and calibrated Hinge Loss and SGD-QN for logistic Loss on Caltech256 and SUN data bases. One sees from the plots that the convergence of our Stochastic Low-Rank Newton approach SLND is a magnitude faster than the classical SGD. Note that accuracy of Logistic Loss and calibrated Hinge Loss SLND are very similar. Accuracy of SGD-QN is very close to SGD; we get similar results when using only a diagonal approximation of the Hessian matrix in our SLND method.

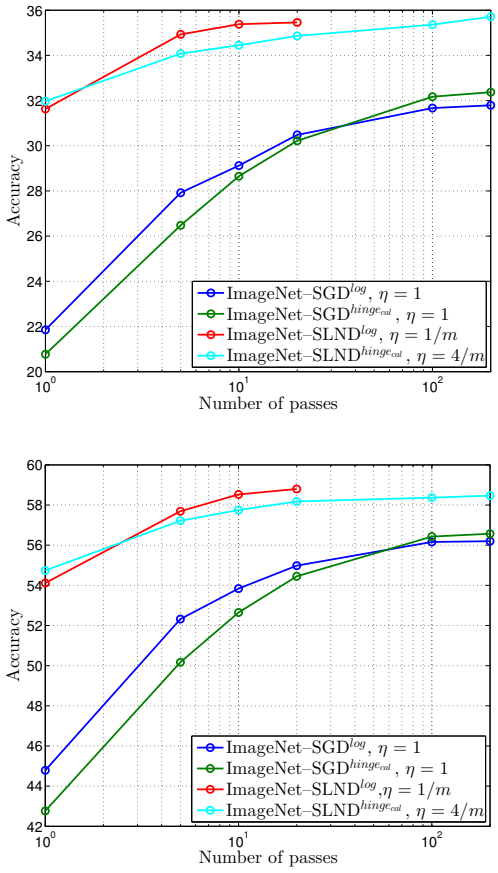

Figure 7: Accuracies as a function of number of passes for SGD and SLND on ImageNet. On top, the top-1 accuracy and at the bottom the top-5 accuracy.

Plots of convergence in Fig 7 on ImageNet shows again that SLND is faster of a magnitude than classical SGD both for the top-1 accuracy and top-5 accu-

racy. Training using SLND with 10 passes on 1,2 millions samples requires only one CPU hours while training SGD requires 20 CPU hours on a 2 X Intel Xeon E5-2687W 3,1GHz and 64 GB of RAM. Thus fast convergence of SLND results in sparse training set requirement well adapted for large scale image classification. Moreover SLND improves significantly accuracy of the SGD baseline.

5 SLND Theoretical convergence analysis

5.1 Best rank k approximation

We first show that $\mathcal{H}^*$, as computed in (8), is the best rank k approximation of the inverse of $\mathcal{H}$ according to squared Frobenius norm.

Lemma 1 *$\mathcal{H}^*$ satisfies:*

$$\mathcal{H}^* = \min_{\mathcal{H}' \in \mathbb{R}^{m \times m}, rank(\mathcal{H}')=k} \|\mathcal{I} - \mathcal{H}\mathcal{H}'\|_F^2 \tag{12}$$

Proof: We use the fact that $\mathcal{H} = \mathcal{P}\mathcal{D}\mathcal{P}^\top$, $\mathcal{P}\mathcal{P}^\top = \mathcal{I}$ and trace tr is cyclic invariant, and we have: $\|\mathcal{I} - \mathcal{H}\mathcal{H}'\|_F^2 = \mathrm{tr}((\mathcal{I} - \mathcal{H}\mathcal{H}')(\mathcal{I} - \mathcal{H}\mathcal{H}')) = \mathrm{tr}(\mathcal{P}\mathcal{P}^\top(\mathcal{I} - \mathcal{H}\mathcal{H}')\mathcal{P}\mathcal{P}^\top(\mathcal{I} - \mathcal{H}\mathcal{H}')) = \mathrm{tr}(\mathcal{P}^\top(\mathcal{I} - \mathcal{H}\mathcal{H}')\mathcal{P}\mathcal{P}^\top(\mathcal{I} - \mathcal{H}\mathcal{H}')\mathcal{P}) = \mathrm{tr}((\mathcal{I} - \mathcal{D}(\mathcal{P}^\top\mathcal{H}'\mathcal{P})))^2)$, out of which is comes that $\mathcal{P}^\top\mathcal{H}'\mathcal{P}$ is diagonal, and so $\mathcal{H}'$ can be diagonalized in the same basis as $\mathcal{H}$. Finally, to minimize the squared Frobenius norm, the non zero entries in its diagonal must equal the k greatest non-zero entries in $\mathcal{D}$. □

5.2 A Weak Separability Assumption

We now prove a convergence result on SLND. For this objective, we define $p_{tj} \doteq -\mathrm{F}'(y_j w_t^\top x_j) \geq 0$ as a weight over the examples. For any classification calibrated loss, $-\mathrm{F}'$ is decreasing. Hence, weight p_{tj} is all the *smaller* as example j is all the *better* classified by w_t. Intuitively, an example gets better classified as y_j agrees with the sign of $w_t^\top x_j$ and the magnitude $|w_t^\top x_j|$ is large. We let $p_t \in \mathbb{R}^m$ be the vector of weights. We let $x_j^\circ \doteq (\mathcal{P}_{|k}\sqrt{\mathcal{D}_{|k}^{-1}})^\top x_j$ denote vector x_j expressed in the normalized eigenvectors' basis of $\mathcal{H}^*$ (8). Finally, we define $s_t \in \mathbb{R}^m$ as the vector whose coordinates are:

$$s_{tj} \doteq y_j x_j^\top \mathcal{H}^* x_{i_t} = y_j (x_j^\circ)^\top x_{i_t}^\circ \ , \forall j \ , \tag{13}$$

where example i_t is the one chosen to update w_t in (11).

Our result relies on the following *Weak Separability Assumption*:

- (WSA) There exists $\gamma > 0$ a constant such that for any iteration t in SLND,

$$\frac{p_t^\top s_t}{\|s_t\|_1} \quad \geq \quad \gamma \ . \tag{14}$$

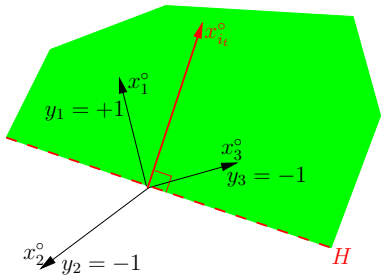

Figure 8: $x^\circ_{i_t}$ is a better classifier for examples 1 and 2 ($s_{t1}, s_{t2} > 0$) than it is for example 3 ($s_{t3} < 0$).

To interpret WSA and see why it is indeed a Weak Separability Assumption, consider the interpretation of s_t and assume $x^\circ_{i_t}$ is used as a linear classifier. Then, $s_{tj} \geq 0$ iff the class y_j agrees with the sign of this classifier, and it is all the larger as the classifier's output is large. On the other hand, $s_{tj} \leq 0$ iff the class y_j disagrees with the sign of the classifier, and it is all the smaller as the classifier's output is large. Hence, s_{tj} quantifies the goodness of fit of classifier $x^\circ_{i_t}$ on x_j (see Figure 8). Thus, $p_t^\top s_t$ is a weighted average of this goodness of fit, in which weights are all the larger as examples have received a bad fitting so far by w_t. Hence, WSA implies that x_{i_t} must contribute to classify better at least a small fraction of the examples with respect to w_t. To see why it is "Weak", informally, picking x_{i_t} at random in any set satisfying mild constraints would make an expected value of $p_t^\top s_t$ equal to zero. So, we require the choice of x_{i_t} in SLND to beat a random linear classifier by at least a small amount. For the informed reader, the WSA parallels in our setting the popular weak learning assumptions in boosting algorithms [7].

5.3 Convergence theorem

The following Theorem shows that under the WSA, there exists a guaranteed decrease rate of the calibrated risk at each iteration, and this holds for whichever of the logistic and calibrated Hinge loss chosen to run SLND. The result would also

hold for various other possible choices of classification calibrated losse, including the squared loss.

Theorem 2 *Assume WSA is satisfied at each step of* SLND. *Then, for any* $\varepsilon \in (0,1)$ *there exists a value of* η_t *in* $\Omega(1/m)$ *and* $O(1/\sqrt{m})$ *such that the following rate of decrease is guaranteed for the calibrated risk at hand:*

$$\varepsilon_{\mathrm{F}}(w_{t+1}, \mathcal{S}) \quad \leq \quad \varepsilon_{\mathrm{F}}(w_t, \mathcal{S}) - \frac{2\gamma^2\epsilon(1-\epsilon)}{m\mathrm{F}''(0)} \ , \forall t \ . \tag{15}$$

Since SLND is initialized with $w_0 = 0$, the null vector, to guarantee $\varepsilon_{\mathrm{F}}(w_T, \mathcal{S}) \leq \mathrm{F}^\circ$ for any chosen real $\mathrm{F}^\circ \leq \mathrm{F}(0)$ such that F° is in the image of F, it is enough to make

$$T \quad \geq \quad \frac{(\mathrm{F}(0) - \mathrm{F}^\circ)\mathrm{F}''(0)}{2\gamma^2\epsilon(1-\epsilon)} \times m = \Omega\left(\frac{m}{\gamma^2}\right)$$

iterations of SLND. In order not to laden the paper's body, a proofsketch of the Theorem is provided in an Appendix. The proof exhibits and discusses the expression of η_t which guarantees (15).

6 Conclusion

In this paper we have proposed a new Stochastic Low Rank Newton descent algorithm (SLND) for the minimization of calibrated risk with linear complexity both in term number of samples and dimension of the features. SLND performs update of the current classifier with pseudo-inverses of the Hessian that are the most accurate low-rank approximations of the inverse according to Frobenius norm. We show the convergence of SLND using a Weak Separability Assumption which states that each example chosen to update the classifier must provide a weighted margin at least larger than some (possibly small) constant $\gamma > 0$. Under this weak assumption, SLND guarantees that its classifier has reached some fixed upper-bound on the claibrated risk at hand after $\Omega(m/\gamma^2)$ iterations. No convergence rates are known to date for SGD-like approaches. Furthermore, the theory provides us with a set of working parameters for the experiments, including a step parameter η_t typically in the order $\Omega(1/m)$.

We validate these theoretical properties by benchmarking it against state-of-the-art SGD algorithm on three challenging domains: Caltech256, SUN and ImageNet. The results on large scale image classification display that SLND improves significantly accuracy of the SGD baseline while being faster by orders of magnitude. Experiments also display that the parameters of SLND may be easily fixed and used from a domain onto another.

References

[1] S.-I. Amari. Natural Gradient works efficiently in Learning. *Neural Computation*, 10:251–276, 1998. 2, 7

[2] P. Bartlett, M. Jordan, and J. D. McAuliffe. Convexity, classification, and risk bounds. *JASA*, 101:138–156, 2006. 4

[3] Antoine Bordes, Léon Bottou, and Patrick Gallinari. Sgd-qn: Careful quasi-newton stochastic gradient descent. *J. Mach. Learn. Res.*, 10:1737–1754, December 2009. 3, 4, 6, 7

[4] Antoine Bordes, Léon Bottou, Patrick Gallinari, Jonathan Chang, and S. Alex Smith. Erratum: Sgdqn is less careful than expected. *J. Mach. Learn. Res.*, 11:2229–2240, August 2010. 3

[5] Lon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In *NIPS*20*, pages 161–168, 2008. 2, 7

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR'09*, 2009. 8

[7] Y. Freund and R. E. Schapire. A Decision-Theoretic generalization of on-line learning and an application to Boosting. *JCSS*, 55:119–139, 1997. 14

[8] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007. 8

[9] Thorsten Joachims. Optimizing search engines using clickthrough data. In *KDD '02*, pages 133–142, New York, NY, USA, 2002. ACM. 2

[10] S. Kakade, S. Shalev-Shwartz, and A. Tewari. Applications of strong convexity–strong smoothness duality to learning with matrices. Technical Report CoRR abs/0910.0610, CoRR, 2009. 18

[11] S Sathiya Keerthi, Olivier Chapelle, Dennis DeCoste, and P Bennett. Building support vector machines with reduced classifier complexity. *JMLR*, 7(7):1493–1515, 2006. 2

[12] Thomas Mensink, Jakob Verbeek, Florent Perronnin, and Gabriela Csurka. Metric Learning for Large Scale Image Classification: Generalizing to New Classes at Near-Zero Cost. In *Procs of the 12 th ECCV*, 2012. 8

[13] E. Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35:773–782, 1980. 7

[14] R. Nock and F. Nielsen. On the efficient minimization of classification-calibrated surrogates. In *NIPS*21*, pages 1201–1208, 2008. 4, 18

[15] F. Perronnin, Z. Akata, Z. Harchaoui, and C. Schmid. Towards good practice in large-scale learning for image classification. In *CVPR'12*, pages 3482–3489, June. 4, 6, 7

[16] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Procs of the 11 th ECCV*, pages 143–156, 2010. 2, 8

[17] Nicol N. Schraudolph, Jin Yu, and Simon Günter. A Stochastic Quasi-Newton Method for Online Convex Optimization. In *AISTATS'07*, pages 436–443, San Juan, Puerto Rico, 2007. JMLR. 2, 7

[18] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *ICML '07*, pages 807–814, New York, NY, USA, 2007. ACM. 2, 7

[19] V. Vapnik. *Statistical Learning Theory*. John Wiley, 1998. 2

[20] E. Vernet, R.-C. Williamson, and M. Reid. Composite multiclass losses. In *NIPS*24*, pages 1224–1232, 2011. 4

[21] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, T. Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *CVPR'10*, pages 3360–3367, 2010. 2

[22] Jason Weston, Samy Bengio, and Nicolas Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *IJCAI'11*, pages 2764–2770, 2011. 4

[23] J. Xiao, J. Hays, K.-A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR'10*, pages 3485–3492, 2010. 8

[24] Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *ICML '04*, pages 116–, New York, NY, USA, 2004. ACM. 2, 7

7 Appendix : proofsketch of Theorem 2

In order to make it fit into the paper, we sketch the proof of the Theorem. To make it more readable, the proofsketch is partitioned into blocks starting by symbol "•". The proofsketch of Theorem 2 involves there steps: • Bregman divergence estimation • Calibrated loss properties • Weak Separability Assumption.

We first make some simplifications in notations. We remove the c subscript and make the analysis for class c, and thus focus on the analysis of $\varepsilon_{\mathrm{F}}(h_c, \mathcal{S})$, noted for short $\varepsilon_{\mathrm{F}}(h, \mathcal{S})$. To avoid confusion, we also rename example chosen at iteration t in (11) as example i_t, so that (11) reads:

$$w_{t+1} = w_t - \eta_t y_{i_t} F'\left(y_{i_t} w_t^T x_{i_t}\right) x_{i_t}^{\circ} \quad . \tag{16}$$

• Bregman divergence estimation: let us define the Legendre conjugate and the notion of Bregman divergence. $\tilde{\mathrm{F}}(x) \doteq \mathrm{F}^{\star}(-x)$, where $\star$ denotes the Legendre conjugate ($\mathrm{F}^{\star}(x) \doteq x(\mathrm{F}')^{-1}(x) - \mathrm{F}((\mathrm{F}')^{-1}(x))$), and $D_{\tilde{\mathrm{F}}}(u \| v) \doteq \tilde{\mathrm{F}}(u) - \tilde{\mathrm{F}}(v) -$

$(u-v)\tilde{\mathrm{F}}'(v)$ denotes the Bregman divergence with generator $\tilde{\mathrm{F}}$ [14].

We get the following equality

$$\begin{aligned}
&\varepsilon_{\mathrm{F}}(w_{t+1},\mathcal{S})-\varepsilon_{\mathrm{F}}(w_t,\mathcal{S})\\
&= \frac{1}{m}\sum_{i=1}^m \mathrm{F}(y_{ic}w_{t+1}^\top x_i)-\frac{1}{m}\sum_{i=1}^m \mathrm{F}(y_{ic}w_t^\top x_i)\\
&= -\frac{1}{m}\sum_{i=1}^m D_{\tilde{\mathrm{F}}}(p_{(t+1)i}\|p_{ti})\\
&\quad -\frac{\eta_t}{m}\sum_{i=1}^m p_{(t+1)i}y_iy_{i_t}\pi(i_t,i)\ , \qquad (17)
\end{aligned}$$

where

$$\pi(i,i_t) \doteq p_{ti_t}x_{i_t}^\top\mathcal{H}^\star x_i = p_{ti_t}(x_i^\circ)^\top x_{i_t}^\circ\ , \qquad (18)$$

• Calibrated loss properties: since $\mathrm{F}''(x)\leq\mathrm{F}''(0)$ for the classification calibrated losses we consider, we also have the following quadratic lower-bound which can be obtained following [10]:

$$\sum_{i=1}^m D_{\tilde{\mathrm{F}}}(p_{(t+1)i}\|p_{ti}) \geq \frac{1}{2\mathrm{F}''(0)}\sum_{i=1}^m (p_{(t+1)i}-p_{ti})^2\ . \qquad (19)$$

Cauchy-Schwartz inequality brings:

$$\begin{aligned}
&\sum_{i=1}^m (y_iy_{i_t}\pi(i_t,i))^2\sum_{i=1}^m (p_{(t+1)i}-p_{ti})^2 \qquad (20)\\
&\geq \left(\sum_{i=1}^m y_iy_{i_t}\pi(i_t,i)(p_{(t+1)i}-p_{ti})\right)^2\ . \qquad (21)
\end{aligned}$$

Define for short $v_t \doteq \sum_{i=1}^m p_{(t+1)i}y_iy_{i_t}\pi(i_t,i)$, $e_t \doteq \sum_{i=1}^m p_{ti}y_iy_{i_t}\pi(i_t,i)$ and $\Pi_t \doteq \sum_{i=1}^m \pi^2(i_t,i)$. Plugging (19) and (21) into (17) and simplifying, we obtain:

$$\begin{aligned}
&\varepsilon_{\mathrm{F}}(w_{t+1},\mathcal{S})-\varepsilon_{\mathrm{F}}(w_t,\mathcal{S})\\
&\leq \underbrace{-\frac{(v_t-e_t)^2}{2\mathrm{F}''(0)m\Pi_t}-\frac{\eta_t v_t}{m}}_{\doteq\frac{\Delta_t(v_t)}{m}}\ . \qquad (22)
\end{aligned}$$

• $\Delta_t(v_t)$ takes its maximum for $v_t = v^\circ = e_t - \mathrm{F}''(0)\eta_t \sum_{i=1}^m (y_i y_{i_t} \pi(i_t, i))^2 = e_t - \mathrm{F}''(0)\eta_t \Pi_t$, for which we have:

$$\Delta_t(v^\circ) \quad = \quad \frac{\mathrm{F}''(0)\eta_t \Pi_t}{2} \times \left(\eta_t - \frac{2e_t}{\mathrm{F}''(0)\Pi_t} \right) \ .$$

Assume we pick, for some $\epsilon \in (0, 1)$:

$$\eta_t \quad \doteq \quad \frac{2(1-\epsilon)e_t}{\mathrm{F}''(0)\Pi_t} \ . \tag{23}$$

For this choice of η_t, we have:

$$\Delta_t(v^\circ) \quad = \quad -\frac{2\epsilon(1-\epsilon)}{\mathrm{F}''(0)} \rho(i_t, \mathcal{H}^*) \ , \tag{24}$$

with

$$\rho(i_t, \mathcal{H}^*) \quad \doteq \quad \frac{\left(\sum_{i=1}^m p_{ti} y_i (x_i^\circ)^\top x_{i_t}^\circ \right)^2}{\sum_{i=1}^m ((x_i^\circ)^\top x_{i_t}^\circ)^2} \ .$$

• Weak Separability Assumption: Now, the Weak Separability Assumption implies $|\sum_{i=1}^m p_{ti} y_i (x_i^\circ)^\top x_{i_t}^\circ| \geq \gamma \|s_t\|_1 \geq \gamma \|s_t\|_2 = \gamma \sqrt{\sum_{i=1}^m ((x_i^\circ)^\top x_{i_t}^\circ)^2}$, which leads to $\rho(i_t, \mathcal{H}^*) \geq \gamma^2$.
Finally, the fact that $\Delta_t(v_t) \leq \Delta_t(v^\circ)$ and (24) imply:

$$\Delta_t(v_t) \quad \leq \quad -\frac{2\gamma^2\epsilon(1-\epsilon)}{\mathrm{F}''(0)} \ .$$

Plugging this into (22) achieves the proof of the theorem.
Remarks on η_t (23) gives, under the WSA:

$$\begin{aligned} \eta_t \quad &= \quad \frac{2(1-\epsilon) \sum_{i=1}^m p_{ti} y_i y_{i_t} \pi(i_t, i)}{\mathrm{F}''(0)\Pi_t} \\ &= \quad \frac{2(1-\epsilon)\gamma' \|s_t\|_1}{\mathrm{F}''(0)|p_{ti_t} y_{i_t}| \|s_t\|_2^2} \ , \end{aligned}$$

for some $\gamma' \geq \gamma > 0$ as in the WSA. Because $\|s_t\|_2 \leq \|s_t\|_1 \leq \sqrt{m}\|s_t\|_2$, it comes:

$$\frac{2(1-\epsilon)\gamma'}{\mathrm{F}''(0)p_{ti_t}\|s_t\|_1} \quad \leq \eta_t \leq \quad \frac{2(1-\epsilon)\gamma'\sqrt{m}}{\mathrm{F}''(0)p_{ti_t}\|s_t\|_1} \ .$$

Letting $\mu_t \doteq (1/m)\sum_{i=1}^{m} |(x_i^\circ)^\top x_{i_t}^\circ|$ denote the average value of $|s_{tj}|$, we obtain:

$$\frac{2(1-\epsilon)\gamma'}{m\mathrm{F}''(0)p_{ti_t}\mu_t} \leq \eta_t \leq \frac{2(1-\epsilon)\gamma'}{\sqrt{m}\mathrm{F}''(0)p_{ti_t}\mu_t} .$$

Hence, omitting p_{ti_t} in big-Oh notations to simplify the analysis, the value η_t which guarantees the rate of convergence of Theorem 2 is indeed roughly between $\Omega(1/m)$ and $O(1/\sqrt{m})$.